

## Claims

[c1] 1. A server comprising:  
at least one port driver, each driver corresponding to a port to which a port device can be connected; and,  
an automatic plug-and-play component designed to detect connection and disconnection of a port device to a port having a corresponding driver selected from the at least one port driver, the component without user intervention automatically installing an appropriate driver for the port device upon connection thereof to the port, such that the device is accessible by clients communicatively coupled to the server, and automatically uninstalling the appropriate driver upon disconnection of the port device from the port, such that the port device is inaccessible by the clients.

[c2] 2. The server of claim 1, wherein the port device is one of: a printer, a scanner, a fax machine, a multi-function device (MFD), and a digital camera.

[c3] 3. The server of claim 1, wherein the port is a parallel port.

[c4] 4. The server of claim 1, wherein the port is a serial port.

[c5] 5. The server of claim 4, wherein the serial port has a Universal Serial Bus (USB) form factor.

[c6] 6. The server of claim 4, wherein the serial port has an IEEE1394 form factor.

[c7] 7. The server of claim 1, wherein upon connection of the port device to the port the automatic plug-and-play component retrieves a plug-and-play identifier from the port device, and selects the appropriate driver based on the plug-and-play identifier.

[c8] 8. The server of claim 1, wherein installing the appropriate driver for the port device includes downloading the appropriate driver from the Internet.

[c9] 9. The server of claim 1, wherein the server is a server appliance lacking at least a dedicated keyboard and a dedicated monitor.

[c10] 10.A machine-readable medium having stored thereon a computer program for execution by a processor of a server communicatively coupled to one or more clients, the computer program comprising:  
at least one port class driver, each port class driver designed to pass signals to and receive signals from a corresponding port driver corresponding to a port to which a port device can be connected; and,  
a monitor designed to monitor connection and disconnection of a port device to a port selected from the at least one port driver, the logic causing without user intervention automatic installation of an appropriate driver for the port device upon connection thereof to the port, such that the device is accessible by the clients, and automatic uninstallation of the appropriate driver upon disconnection of the port device from the port, such that the port device is inaccessible by the clients.

[c11] 11.The medium of claim 10, wherein the port device is one of: a printer, a scanner, a fax machine, a multi-function device (MFD), and a digital camera.

[c12] 12.The medium of claim 10, wherein the port is a parallel port.

[c13] 13.The medium of claim 10, wherein the port is a serial port.

[c14] 14.The medium of claim 10, wherein the monitor is implemented as a service.

[c15] 15.The medium of claim 10, wherein the monitor resides in a user mode, and each of the at least one port class driver resides in a kernel mode.

[c16] 16.The medium of claim 10, wherein the monitor incorporates monitoring logic implemented as a state-transition system.

[c17] 17.The medium of claim 10, wherein upon connection of the port device to the port the monitor retrieves a plug-and-play identifier from the port device, and selects the appropriate driver based on the plug-and-play identifier.

[c18] 18.The medium of claim 10, wherein the program further comprises a

support component designed to pass signals from the monitor to the at least one port class driver and vice-versa.

[c19] 19.The medium of claim 18, wherein the support program resides in a user mode.

[c20] 20.The medium of claim 18, wherein the support component comprises a first component and at least one second component, the first component designed to determine a number of ports of the server and instantiate a number of the at least one second component equal to the number of ports, each second component designed to pass signals from the monitor to one of the at least one port class driver and vice-versa.

[c21] 21.The medium of claim 18, wherein each of the first component and the at least one second component is implemented as an object.

[c22] 22.The medium of claim 18, wherein the support component is able to pass signals from an external monitor not residing at the server to the at least one port class driver and vice-versa.

[c23] 23.The medium of claim 22, wherein the external monitor is implemented as a web component.

[c24] 24.A machine-readable medium having stored thereon as a computer program for execution by a processor a monitoring logic for automatic device plug-and-play without user intervention that is implemented as a state transition system comprising:  
a first state in which a device driver is not installed and a plug-and-play identifier has been detected from a device connected to a port;  
a second state in which the device driver is installed and the plug-and-play identifier has been detected from the device connected to the port;  
a third state in which the device driver is not installed and the plug-and-play identifier has not been detected from the device connected to the port; and,  
a fourth state in which the device driver is installed and the plug-and-play identifier has not been detected from the device connected to the port,

wherein the states are transitioned among one another based on a set of transitions comprising:

a first state-to-second state transition based on a manual driver installation event;

a first state-to-third state transition based on an event of no detection of the identifier;

a second state-to-first state transition based on a manual driver uninstallation event;

a second state-to-second state transition based on an event of detection of a new identifier;

a second state-to-fourth state transition based on the event of no detection of the identifier;

a third state-to-second state transition based on an event of detection of the identifier;

a third state-to-fourth state transition based on the manual driver installation event;

a fourth state-to-third state transition based on the manual driver uninstallation event; and,

a fourth state-to-second state transition based on the event of detecting the identifier.